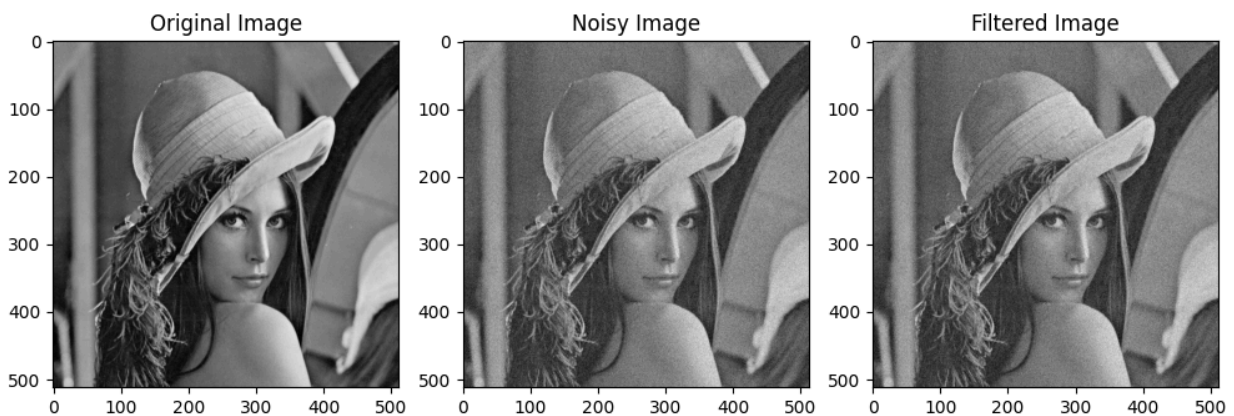


Project Report: Exposure Value Mean Filtering

This project focused on developing a rank order EV (Expected Value) filter aimed at reducing noise in grayscale images while preserving essential image details. Using a mean-based approach, the filter calculates the average of neighboring pixel values within a specified window, effectively smoothing random noise across the image. The filtering process is guided by the standard deviation of the noisy image, which serves as the EV parameter. This parameter allows the filter to adjust dynamically: images with higher noise levels receive a stronger noise-suppressing effect, while lower-noise images are processed more subtly, balancing noise reduction with detail retention. To handle image borders consistently, the function employs mirrored padding, where edges are reflected to create a seamless transition. This padding method prevents boundary distortions commonly encountered in filtering, enabling the filter to operate uniformly across all pixels without introducing artifacts.

RMSE: 12.19 %, PSNR: 26.41 dB



Noise fraction = 0.3

When running the code this is the output in the terminal. We begin by choosing the image you want to process, in this case `lena_y` - > but in the `main.py` the image is then converted into grayscale. The window size is 3 and the `rmse` ~ 12.19 % and `psnr` ~ 26.41 dB. The `rmse` value is considered good as it is considered to closely resemble the original. The **Original vs. Filtered metrics** show that the filter preserves the original image's quality well (low RMSE, high PSNR). Compared to my first submission I was rushing and did the whole thing wrong because of issues outside of class and this assignment. This assignment I was ashamed of what I did and hope this is the proper implementation of the code. I didn't realize that even from the beginning it wasn't calling the file and that's why Gaussian Noise wasn't added and the whole thing was a mess. I did look on the internet and that was a great mistake.

Available TIFF files from Testing directory:

1: Girl.tif
2: Airplane-F16.tif
3: Bridge.tif
4: Barbara.tif
5: Lena_Y.tif
6: Lake_Y.tif
7: SailBoat.tif
8: Boat.tif
9: Lena_Y-Scratches.tif
10: BigBen.tif
11: Lena_Color.tif
12: LowContrast-a.tif
13: Temple_Y.tif
14: LowContrast-b.tif
15: Pepper_Y.tif
16: Airplane-F16_Y.tif
17: LowContrast-c.tif
18: Pepper.tif

Choose the first file (1-18): 5

Processing files: ./Test_Images/Lena_Y.tif

Input the noise fraction for image: 0.3

Input the window size: 3

Original v. Filtered	
RMSE	6.843373102998957 %
PSNR	31.425399238224156 dB

Noisy v. Filtered	
RMSE	12.191560099973021 %
PSNR	26.40961793034498 dB

2024-12-06 22:23:41.704 Python[37086:722360] +[IMKClient subclass]: chose IMKClient_Modern

2024-12-06 22:23:41.704 Python[37086:722360] +[IMKInputSession subclass]: chose IMKInputSession_Modern

Do you want to process another pair of files? (Y/N): N